THALES

OpenShift Container Platform

INTEGRATION GUIDE

THALES LUNA HSM & DPOD LUNA CLOUD HSM

Document Information

Document Part Number	007-000821-001
Revision	A
Release Date	7 December 2020

Trademarks, Copyrights, and Third-Party Software

Copyright © 2020 Thales Group. All rights reserved. Thales and the Thales logo are trademarks and service marks of Thales Group and/or its subsidiaries and are registered in certain countries. All other trademarks and service marks, whether registered or not in specific countries, are the property of their respective owners.

CONTENTS

Copyright © 2020 Thales Group

Overview

OpenShift Container Platform brings together Docker and Kubernetes, and provides an API to manage these services. OpenShift Container Platform allows you to create and manage containers. Containers are standalone processes that run within their own environment, independent of the operating system and the underlying infrastructure. OpenShift helps you develop, deploy, and manage container-based applications. It provides you with a self-service platform to create, modify, and deploy applications on demand, thus enabling faster development and release life cycles.

Luna HSMs are Hardware Security Module (HSM) devices for storage of keys and managing cryptographic operations to secure container based applications. Following are some of the benefits of using Luna HSMs along with OpenShift container-based applications:

- Secure generation, storage, and protection of cryptographic keys on FIPS 140-2 level 3 validated hardware.
- Full life cycle management of keys.
- HSM audit trail.
- Significant performance improvements by off-loading cryptographic operations from servers.
- Using Cloud services with confidence.

*Cloud HSM services do not have access to the secure audit trail.

Certified Platforms

This integration is tested on the following platforms:

HSM Type	Platforms Tested	OpenShift Version
Luna HSM	CentOS 7	3.11
Luna Cloud HSM	CentOS 7	3.11

Luna HSM: Luna HSM appliances are purposefully designed to provide a balance of security, high performance, and usability that makes them an ideal choice for enterprise, financial, and government organizations. Luna HSMs physically and logically secure cryptographic keys and accelerate cryptographic processing. The Luna HSM on premise offerings include the Luna Network HSM, Luna PCIe HSM, and Luna USB HSMs. Luna HSMs are also available for access as an offering from cloud service providers such as IBM cloud HSM, and AWS cloud HSM classic.

Luna Cloud HSM: Luna Cloud HSM is a cloud-based platform that provides a Cloud HSM service for your organization’s cryptographic operations on DPoD. Using Luna Cloud HSM service security is simple, cost effective and easy to manage because there is no hardware to buy, deploy and maintain. As an Application Owner, you click and deploy services, generate usage reports and maintain just the services you need.

Copyright © 2020 Thales Group

Prerequisites

Before you proceed with the integration, complete the following tasks:

Configuring Luna HSM

To configure Luna HSM:

1. Ensure that the HSM is set up, initialized, provisioned and ready for deployment. Refer to the HSM product documentation for help.
2. Create a partition that will be later used by OpenShift Container Platform.
3. Create and exchange certificate between the Luna Network HSM and Client system. Register client and assign partition to create an NTLS connection. Initialize Crypto Officer and Crypto User roles for the registered partition.
4. Ensure that the partition is successfully registered and configured. The command to see the registered partitions is:

```
# /usr/safenet/lunaclient/bin/lunacm.exe
lunacm (64-bit) v10.2.0-111. Copyright (c) 2020 SafeNet. All rights reserved.
   Available HSMs:
   Slot Id ->              0
   Label ->                OSPod01
   Serial Number ->        1238696044915
   Model ->                LunaSA 7.4.0
   Firmware Version ->     7.4.0
   Configuration ->        Luna User Partition With SO (PW) Signing With
   Cloning Mode
   Slot Description ->     Net Token Slot
   FM HW Status ->         Non-FM
   Current Slot Id: 0
```

5. For PED-authenticated HSM, enable partition policies 22 and 23 to allow activation and auto-activation.

> **NOTE:** Refer to Luna HSM documentation for detailed steps on creating NTLS connection, initializing the partitions, and assigning various user roles.

To use Luna HSM in FIPS Mode

Under FIPS 186-3/4, the RSA methods permitted for generating keys are 186-3 with primes and 186-3 with aux primes. RSA PKCS and X9.31 key generation are no longer approved for operation in a FIPS-compliant HSM. If you are using Luna HSM in FIPS mode, make the following change in configuration file:

```
[Misc]
RSAKeyGenMechRemap=1
```

This setting redirects the older calling mechanism to a new mechanism when Luna HSM is in FIPS mode.

Copyright © 2020 Thales Group

NOTE: For Luna Client 10.x onwards, this setting is not needed. It is applicable for Luna Client 7.x only.

To configure Luna HSM HA (High-Availability)

Please refer to the Luna HSM documentation for HA steps and details regarding configuring and setting up two or more HSM appliances on Windows and UNIX systems. You must enable the HAOnly setting in HA for failover to work so that if primary stop functioning for some reason, all calls automatically routed to secondary till primary starts functioning again.

NOTE: This integration is tested in both HA and FIPS mode.

Configure Luna Cloud HSM Service

You can configure Luna Cloud HSM Service in the following ways:

> Standalone Cloud HSM service using minimum client package

> Standalone Cloud HSM service using full Luna client package

> Luna HSM and Luna Cloud HSM service in hybrid mode

NOTE: Luna Client v10.x or higher is required for configuring Luna HSM device and Luna Cloud HSM service in hybrid mode.

Standalone Cloud HSM service using minimum client package

To configure Luna Cloud HSM service using minimum client package:

1. Transfer the downloaded .zip file to your Client workstation using pscp, scp, or other secure means.
2. Extract the .zip file into a directory on your client workstation.
3. Extract or untar the appropriate client package for your operating system. Do not extract to a new subdirectory; place the files in the client install directory.

   ```
   [Windows]
   cvclient-min.zip
   [Linux]
   cvclient-min.tar
   # tar -xvf cvclient-min.tar
   ```

4. Run the setenv script to create a new configuration file containing information required by the Luna Cloud HSM service.

   ```
   [Windows]
   Right-click setenv.cmd and select Run as Administrator.
   [Linux]
   Source the setenv script.
   # source ./setenv
   ```

5. Run the LunaCM utility and verify the Cloud HSM service is listed.

Copyright © 2020 Thales Group

Standalone Cloud HSM service using full Luna client package

To configure Luna Cloud HSM service using full Luna client package:

1. Transfer the downloaded .zip file to your Client workstation using pscp, scp, or other secure means.
2. Extract the .zip file into a directory on your client workstation.
3. Extract or untar the appropriate client package for your operating system. Do not extract to a new subdirectory; place the files in the client install directory.

   ```
   [Windows]
   cvclient-min.zip
   [Linux]
   cvclient-min.tar
   # tar -xvf cvclient-min.tar
   ```

4. Run the setenv script to create a new configuration file containing information required by the Luna Cloud HSM service.

   ```
   [Windows]
   Right-click setenv.cmd and select Run as Administrator.
   [Linux]
   Source the setenv script.
   # source ./setenv
   ```

5. Copy the server and partition certificates from the Cloud HSM service client directory to Luna client certificates directory:

   Cloud HSM Certificates:

   ```
   server-certificate.pem
   partition-ca-certificate.pem
   partition-certificate.pem
   ```

   LunaClient Certificate Directory:

   ```
   [Windows default location for Luna Client]
   C:\Program Files\Safenet\Lunaclient\cert\
   [Linux default location for Luna Client]
   /usr/safenet/lunaclient/cert/
   ```

   NOTE: Skip this step for Luna Client v10.2 or higher.

6. Open the configuration file from the Cloud HSM service client directory and copy the XTC and REST section.

   ```
   [Windows]
   crystoki.ini
   [Linux]
   Chrystoki.conf
   ```

Copyright © 2020 Thales Group

7. Edit the Luna Client configuration file and add the XTC and REST sections copied from Cloud HSM service client configuration file.
8. Change server and partition certificates path from step 5 in XTC and REST sections. Do not change any other entries provided in these sections.

   [XTC]

```
. . .
PartitionCAPath=<LunaClient_cert_directory>\partition-ca-certificate.pem
PartitionCertPath00=<LunaClient_cert_directory>\partition-certificate.pem
. . .

[REST]
. . .
SSLClientSideVerifyFile=<LunaClient_cert_directory>\server-certificate.pem
. . .
```

   NOTE: Skip this step for Luna Client v10.2 or higher.

9. Edit the following entry from the Misc section and update the correct path for the plugins directory:

```
Misc]
PluginModuleDir=<LunaClient_plugins_directory>

[Windows Default]

C:\Program Files\Safenet\Lunaclient\plugins\

[Linux Default]

/usr/safenet/lunaclient/plugins/
```

   Save the configuration file. If you wish, you can now safely delete the extracted Cloud HSM service client directory.

10. Reset the ChrystokiConfigurationPath environment variable and point back to the location of the Luna Client configuration file.

    [Windows]

    In the Control Panel, search for "environment" and select Edit the system environment variables. Click Environment Variables. In both list boxes for the current user and system variables, edit ChrystokiConfigurationPath and point to the crystoki.ini file in the Luna client install directory.

    [Linux]

    Either open a new shell session or export the environment variable for the current session pointing to the location of the Chrystoki.conf file:

```
# export ChrystokiConfigurationPath=/etc/
```

11. Run the LunaCM utility and verify that the Cloud HSM service is listed. Both Luna and Cloud HSM service will be listed in the hybrid mode.

    NOTE: Follow the Luna Cloud HSM documentation for detailed steps for creating service, client, and initializing various user roles.

Copyright © 2020 Thales Group

Luna HSM and Luna Cloud HSM service in hybrid mode

To configure Luna HSM and Luna Cloud HSM service in hybrid mode, follow the steps mentioned under the Standalone Cloud HSM service using full Luna client package section above.

> **NOTE:** Luna Client v10.x or higher is required for configuring Luna HSM device and Luna Cloud HSM service in hybrid mode.

To use Luna Cloud HSM Service in FIPS mode

Cloud HSM service operates in both FIPS and non-FIPS mode. If your organization requires non-FIPS algorithms for your operations, enable the Allow non-FIPS approved algorithms check box when configuring your Cloud HSM service. The FIPS mode is enabled by default. Refer to the Mechanism List in the SDK Reference Guide for more information about available FIPS and non-FIPS algorithms.

Set up OpenShift Container Platform

Refer to OpenShift Documentation for installing and running the OpenShift Container Platform. For demonstration, OpenShift Container Platform used in this documentation is setup with 1 Master node on VMware. After installation, ensure that OpenShift Container Platform is up and running successfully.

```
[root@NOIHSM1INT-OSO bin]# oc login -u system:admin
Logged into "https://127.0.0.1:8443" as "system:admin" using existing credentials.

You have access to the following projects and can switch between them with 'oc project <projectname>':

    default
    kube-dns
    kube-proxy
    kube-public
    kube-system
  * myproject
    openshift
    openshift-apiserver
    openshift-controller-manager
    openshift-core-operators
    openshift-infra
    openshift-node
    openshift-service-cert-signer
    openshift-web-console

Using project "myproject".
[root@NOIHSM1INT-OSO bin]# oc get nodes
NAME        STATUS    ROLES     AGE       VERSION
localhost   Ready     <none>    12d       v1.11.0+d4cacc0
[root@NOIHSM1INT-OSO bin]# oc get pods --all-namespaces
NAMESPACE                       NAME                                                READY     STATUS      RESTARTS   AGE
default                         docker-registry-1-dgcq5                             1/1       Running     0          12d
default                         persistent-volume-setup-9mtd9                       0/1       Completed   0          12d
default                         router-1-hz4x8                                      1/1       Running     1          12d
kube-dns                        kube-dns-d8cf4                                      1/1       Running     0          12d
kube-proxy                      kube-proxy-gpnl6                                    1/1       Running     0          12d
kube-system                     kube-controller-manager-localhost                   1/1       Running     1          12d
kube-system                     kube-scheduler-localhost                            1/1       Running     1          12d
kube-system                     master-api-localhost                                1/1       Running     3          12d
kube-system                     master-etcd-localhost                               1/1       Running     0          12d
openshift-apiserver             openshift-apiserver-99ntk                           1/1       Running     0          12d
openshift-controller-manager    openshift-controller-manager-xmkl2                  1/1       Running     1          12d
openshift-core-operators        openshift-service-cert-signer-operator-6d477f986b-wgvzt   1/1       Running     1          12d
openshift-core-operators        openshift-web-console-operator-664b974ff5-skpxw     1/1       Running     1          12d
openshift-service-cert-signer   apiservice-cabundle-injector-8ffbbb6dc-f9wzs        1/1       Running     0          12d
openshift-service-cert-signer   service-serving-cert-signer-668c45d5f-n5rs2         1/1       Running     0          12d
openshift-web-console           webconsole-5fdb8cc67-2chdr                          1/1       Running     0          12d
[root@NOIHSM1INT-OSO bin]#
```

Copyright © 2020 Thales Group

Configuring Luna HSMs in OpenShift Container

By configuring Luna HSM in an OpenShift Container Platform, you can secure Cryptographic Keys on Luna HSM that can be accessible for any application running in Pod. A pod is one or more containers deployed together on one host, and the smallest compute unit that can be defined, deployed, and managed. Each pod is allocated its own internal IP address, and containers within pods can share their local storage and networking. You can configure Luna HSM inside a pod. Any configuration updates to the OpenShift Master node will be automatically deployed on all nodes connected to the Master.

Configuring Luna HSMs within Pod

To use Luna HSM or Luna Cloud HSM service with OpenShift Container Platform, first you need to create a Docker image containing Luna Client software to communicate with Luna HSM or Luna Cloud HSM Service. You can then use the Docker image to create the Pod that will communicate with Luna HSM or Luna Cloud HSM Service.

1. Connect to the master host as root or as a user with administrative privileges.
2. Set the WORKDIR. Typically, the value will be a directory tree child of your development workspace.

   ```
   # export WORKDIR=/opt/developer/myproject
   # mkdir -p $WORKDIR
   # cd $WORKDIR
   ```

3. Copy the Luna minimal client package at the WORKDIR directory:

   ```
   # cp /home/610-000401-002_SW_Minimal_Client_10.2_Linux_RevA.tar $WORKDIR
   ```

4. Copy the Luna Configuration file /etc/Chrystoki.conf file to WORKDIR directory:

   ```
   # cp /etc/Chrystoki.conf $WORKDIR
   ```

   NOTE: When using the Luna Cloud HSM, copy the configuration file from the location where you unzip the client.

5. Edit the $WORKDIR/Chrystoki.conf and add/update the following sections with the changes mentioned below. Do not update/replace other settings in the configuration file.

```
Chrystoki2 = {
   LibUNIX = /usr/safenet/lunaclient/libs/64/libCryptoki2.so;
   LibUNIX64 = /usr/safenet/lunaclient/libs/64/libCryptoki2_64.so;
}
Misc = {
   PluginModuleDir = /usr/safenet/lunaclient/plugins;
}
Secure Trusted Channel = {
   ClientTokenLib = /usr/safenet/lunaclient/libs/64/libSoftToken.so;
}
```

Copyright © 2020 Thales Group

NOTE: "PluginModuleDir" settings is applicable for the Luna Cloud HSM service and Hybrid mode. "ClientTokenLib" settings is applicable for Luna HSM and Hybrid mode.

6. Copy the certificates and keys required to connect to Luna HSM in $WORKDIR directory:

```
# cp -r /usr/safenet/lunaclient/cert/server $WORKDIR
# cp -r /usr/safenet/lunaclient/cert/client $WORKDIR
```

NOTE: Skip this step for Luna Cloud HSM Service, if using the Luna Client v10.2 or higher.

7. Create a Dockerfile under the $WORKDIR directory that will create a Docker image containing all the required resources for Luna HSM Client to communicate with the Luna HSM or Luna Cloud HSM Service. Ensure that you are providing the correct file name and path for all the required certificates that are copying from host to Docker image. The certificate names and path are different for Luna HSM and Luna Cloud HSM Service.

```
# cat Dockerfile
```

```
### Docker Image
FROM centos:centos7 as luna-client-image
COPY 610-000401-002_SW_Minimal_Client_10.2_Linux_RevA.tar /tmp/

RUN mkdir -p /usr/safenet/lunaclient
RUN mkdir -p /usr/safenet/lunaclient/cert
RUN mkdir -p /usr/safenet/lunaclient/cert/client
RUN mkdir -p /usr/safenet/lunaclient/cert/server

RUN tar -xvf /tmp/610-000401-002_SW_Minimal_Client_10.2_Linux_RevA.tar
--strip 1 -C /usr/safenet/lunaclient
RUN cp /usr/safenet/lunaclient/openssl.cnf /usr/safenet/lunaclient/bin
ENV ChrystokiConfigurationPath=/etc
COPY Chrystoki.conf /etc/Chrystoki.conf

COPY server/CAFile.pem /usr/safenet/lunaclient/cert/server
COPY client/NOIHSM1INT-OSOKey.pem /usr/safenet/lunaclient/cert/client
COPY client/NOIHSM1INT-OSO.pem /usr/safenet/lunaclient/cert/client

ENTRYPOINT ["/bin/bash"]
```

NOTE: Copying the client and server certificates is not applicable if you are using the Luna Cloud HSM Service with Luna Client 10.2 or higher. For lower version of Luna Client, ensure that the certificate names and paths correspond to the Chrystoki.conf configuration file.

Copyright © 2020 Thales Group

8. Create an image with `docker build` command and Dockerfile:

   ```
   # docker build . -t luna-client-image
   ```

   You will see a confirmation message similar to the following when the image is built successfully.

```
LunaClient-Minimal-10.2.0-111.x86_64/bin/64/plink
LunaClient-Minimal-10.2.0-111.x86_64/bin/64/cmu
LunaClient-Minimal-10.2.0-111.x86_64/bin/64/multitoken
LunaClient-Minimal-10.2.0-111.x86_64/bin/64/vtl
LunaClient-Minimal-10.2.0-111.x86_64/008-010068-001_EULA_HSM7_SW_revB.txt
LunaClient-Minimal-10.2.0-111.x86_64/plugins/libdpod.plugin
Removing intermediate container e535de513452
 ---> 5520063ffff9
Step 8/14 : RUN cp /usr/safenet/lunaclient/openssl.cnf /usr/safenet/lunaclient/bin
 ---> Running in c4c8cf1d28f9
Removing intermediate container c4c8cf1d28f9
 ---> 69497c6857b1
Step 9/14 : ENV ChrystokiConfigurationPath=/etc
 ---> Running in bf562e1e5129
Removing intermediate container bf562e1e5129
 ---> a341db826024
Step 10/14 : COPY Chrystoki.conf /etc/Chrystoki.conf
 ---> 59c8c6959491
Step 11/14 : COPY server/CAFile.pem /usr/safenet/lunaclient/cert/server
 ---> 921ec5ddb149
Step 12/14 : COPY client/NOIHSM1INT-OSOKey.pem /usr/safenet/lunaclient/cert/client
 ---> fca2ae2ba516
Step 13/14 : COPY client/NOIHSM1INT-OSO.pem /usr/safenet/lunaclient/cert/client
 ---> b4964c620af7
Step 14/14 : ENTRYPOINT ["/bin/bash"]
 ---> Running in 979aa4cf9900
Removing intermediate container 979aa4cf9900
 ---> 46b2c04791d2
Successfully built 46b2c04791d2
Successfully tagged luna-client-image:latest
[root@NOIHSM1INT-OSO myproject]#
```

9. The image luna-container will be listed along with other images:

   ```
   # docker images
   ```

   The created image will be listed along with other images. The output will be similar to the following:

```
[root@NOIHSM1INT-OSO myproject]# docker images
REPOSITORY                                TAG                 IMAGE ID            CREATED             SIZE
luna-client-image                         latest              46b2c04791d2        3 minutes ago       275MB
centos                                    centos7             8652b9f0cb4c        12 days ago         204MB
openshift/origin-control-plane            v3.11               ac09802e76a2        2 weeks ago         833MB
openshift/origin-hyperkube                v3.11               9a17ce58b661        2 weeks ago         510MB
openshift/origin-hypershift               v3.11               ec0ae2036733        2 weeks ago         550MB
openshift/origin-node                     v3.11               7f0df838304c        2 weeks ago         1.17GB
openshift/origin-control-plane            <none>              81a450d5efc9        2 weeks ago         833MB
openshift/origin-hyperkube                <none>              7bd9c673e6cd        2 weeks ago         510MB
openshift/origin-hypershift               <none>              cdd0b232e854        2 weeks ago         550MB
openshift/origin-deployer                 v3.11               96344491d8c8        2 weeks ago         384MB
openshift/origin-haproxy-router           v3.11               5b277bb0a1c0        2 weeks ago         412MB
openshift/origin-pod                      v3.11               74938d645943        2 weeks ago         262MB
openshift/origin-cli                      v3.11               8d0c65051ed9        2 weeks ago         384MB
openshift/origin-docker-registry          v3.11               9dffb2abf1dd        22 months ago       310MB
openshift/origin-web-console              v3.11               be30b6cce5fa        2 years ago         339MB
openshift/origin-service-serving-cert-signer   v3.11          47dadf9d43b6        2 years ago         276MB
[root@NOIHSM1INT-OSO myproject]#
```

Copyright © 2020 Thales Group

10. Create the pod manifest yaml file luna-config.yaml and ensure that the file has the following contents:

```
# cat luna-config.yaml
```

```
apiVersion: v1
kind: Pod

metadata:
  name: luna-client-pod
  labels:
    openshift.io/name: luna-client-pod

spec:
  hostNetwork: true
  restartPolicy: Always
  containers:
    - name: luna-client-pod
      image: "luna-client-image"
      imagePullPolicy: IfNotPresent

     # Just spin & wait forever
      command: [ "/bin/bash", "-c", "--" ]
      args: [ "while true; do sleep 30; done;" ]
```

NOTE: Remove “hostNetwork: true” setting from YAML, if not using OpenShift privileged service account.

11. Deploy the pod in to the OpenShift Container Platform using luna-config.yaml file.

```
# oc create -f luna-config.yaml
```

```
[root@NOIHSM1INT-OSO myproject]# oc create -f luna-config.yaml
pod/luna-client-pod created
```

12. OpenShift Container Platform will deploy the luna-client-pod. Verify the deployment status:

```
# oc get pods
```

```
[root@NOIHSM1INT-OSO myproject]# oc get pods
NAME              READY     STATUS    RESTARTS   AGE
luna-client-pod   1/1       Running   0          1m
[root@NOIHSM1INT-OSO myproject]#
```

13. Connect the pod to verify that Luna HSM or Luna Cloud HSM Service is available and accessible. Execute the following command on the master host to connect pod:

```
# oc rsh luna-client-pod
```

Copyright © 2020 Thales Group

14. Verify that the pod has access to the Luna HSM partition or Luna Cloud HSM Service:

```
# /usr/safenet/lunaclient/bin/64/lunacm
```

```
[root@NOIHSM1INT-OSO myproject]# oc rsh luna-client-pod
sh-4.2$ /usr/safenet/lunaclient/bin/64/lunacm
lunacm (64-bit) v10.2.0-111. Copyright (c) 2020 SafeNet. All rights reserved.


        Available HSMs:

        Slot Id ->              0
        Label ->                OSPod01
        Serial Number ->        1238696044915
        Model ->                LunaSA 7.4.0
        Firmware Version ->     7.4.0
        Configuration ->        Luna User Partition With SO (PW) Signing With Cloning Mode
        Slot Description ->     Net Token Slot
        FM HW Status ->         Non-FM

        Slot Id ->              1
        Label ->                OSPod02
        Serial Number ->        1238696044951
        Model ->                LunaSA 7.4.0
        Firmware Version ->     7.4.0
        Configuration ->        Luna User Partition With SO (PW) Key Export With Cloning Mode
        Slot Description ->     Net Token Slot
        FM HW Status ->         Non-FM

        Slot Id ->              8
        Label ->                OSCloudPod
        Serial Number ->        1334049518154
        Model ->                Cryptovisor7
        Firmware Version ->     7.3.0
        CV Firmware Version ->  1.4.0
        Configuration ->        Luna User Partition With SO (PW) Signing With Cloning Mode
        Slot Description ->     Net Token Slot
        FM HW Status ->         FM Not Supported

        Slot Id ->              9
        HSM Label ->            OSPodHA
        HSM Serial Number ->    11238696044915
        HSM Model ->            LunaVirtual
        HSM Firmware Version -> 7.4.0
        HSM Configuration ->    Luna Virtual HSM (PW) Signing With Cloning Mode
        HSM Status ->           N/A - HA Group


        Current Slot Id: 0


lunacm:>
```

This completes the integration of OpenShift Container Platform with Luna HSMs. To use Luna HSM or Luna Cloud HSM, run any application in the Pod that supports the HSMs for securing cryptographic keys.

Copyright © 2020 Thales Group

Contacting Customer Support

If you encounter a problem at any stage during this integration, contact Thales Customer Support. Thales Customer Support operates 24 hours a day, 7 days a week. Your level of access to this service is governed by the support plan arrangements made between Thales and your organization. Please consult this support plan for further information about your entitlements, including the hours when telephone support is available to you.

Customer Support Portal

The Customer Support Portal at https://supportportal.thalesgroup.com is a database where you can find solutions for most common problems. The Customer Support Portal is a comprehensive, fully searchable repository of support resources, including software and firmware downloads, release notes listing known problems and workarounds, a knowledge base, FAQs, product documentation, technical notes, and more. You can also use the portal to create and manage support cases.

> **NOTE:** You require an account to access the Customer Support Portal. To create a new account, go to the portal and click on the **REGISTER** link.

Telephone Support

If you have an urgent problem, or cannot access the Customer Support Portal, you can contact Thales Customer Support by telephone at +1 410-931-7520. Additional local telephone support numbers are listed on the support portal.

Email Support

You can also contact technical support by email at technical.support.DIS@thalesgroup.com.

Copyright © 2020 Thales Group